UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/315,334	05/20/1999	MICHAEL E. D. WINSER	40062.12US01	6400

23552 7590 02/07/2005

MERCHANT & GOULD PC
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

EXAMINER	
QUELER, ADAM M	

ART UNIT	PAPER NUMBER
2179	

DATE MAILED: 02/07/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/315,334	WINSER, MICHAEL E. D.	
	Examiner	Art Unit	
	Adam M Queler	2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 4-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 4-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: Amendment filed 10/18/2004, and Remarks filed 07/09/2004.

2. Claims 1 and 4-33 are pending in the case. Claims 1, 5, 14, 19, 24, 25, and 29 are independent claims.

3. The rejection of claim 1 under §102 is withdrawn in view of Applicant's amendment, and newly rejected as obvious as described below.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 4, 19, 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mathews et al. "Vector Markup Language (VML), World Wide Web Consortium Note", published 13-May-1998, and further in view of Harris et al. (USPN 5276607—published on 1/4/1994).

Regarding independent claim 1, Matthews discloses expressions that establish a relationship between properties in code executable by a browser, each property defining a characteristic of an object (p.5, example, <shapetype> tag). Within the <shapetype> contains an attribute "adj." Adj values are the scalar values. Throughout the rest of the entity each "adj" value is represented by the variables from *#0* through *#7*. There also several expressions or formulas within the formulas tag. Each formula is represented throughout the entity by the variables *@0* through

@n. The results of the expressions are used as the properties for various parts in the attribute of the shapetype. Therefore there is a first scalar value and a second property that is a function of the first property. Mathews discloses that editors must include an implementation for supporting the parameterization of values, that is, the changing of values dependent on the scalars (p. 8, #4). Matthews does not specifically discuss the implementation.

Harris discloses automatically changing the second property in response to a change in the first (col. 2, ll. 61-69). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22).

Regarding dependent claim 4, as the Applicant defines the inclusion of non-standard expression additions to HTML and CSS as still being HTML (Specification, p. 11), the VML additions to CSS and HTML are also deemed to be HTML (pp. 4-6). An HTML document is inherently a user interface. The updating of expressions upon change of a scalar was obvious as discussed in claim 2 above.

Regarding dependent claim 19, Mathews teaches scalars and expressions of scalars that are used to program formatting instructions as set forth in claim 1 above. Harris discloses a dependent/dependency relationship between scalar properties and expressions that allows the expressions to be recalculated (col. 1, ll. 20-25). Inherently the code must be analyzed to determine where scalar properties and expressions are. Harris discloses executing an expression dependent on a scalar property upon notification of a change in that property (col. 2, ll. 61-69).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22).

Regarding dependent claim 23, while references are silent as to a monitor, it would have been obvious to one of ordinary skill in the art at the time of the invention to include a monitor for display of the HTML document, as it was a well-known peripheral for computers, for the purpose of displaying data.

Regarding independent claims 24 and 25, the apparatus and medium contain the same limitations as the computer of claim 19 and are rejected for analogous reasons.

6. Claims 5-18, 20-22, and 26-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mathews and further in view of Harris, and further in view of Garman (USPN 5926822—filed on 9/6/1996).

Regarding independent claim 5, Mathews teaches scalars and expressions of scalars that are used to program formatting instructions as set forth in claim 1 above. Mathews does not explicitly teach how the calculations are done. Harris discloses a first object that is a scalar and a second object that is a function of the first (col. 4, ll. 50-59). Harris also discloses a dependent/dependency relationship between scalar properties and expressions that allows the expressions to be recalculated (col. 1, ll. 20-25). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris

specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22). Harris is silent as to using nodal structure with pointers. Gorman discloses a tree for doing such recalculations (col. 9, ll. 3-5). Trees were well known to contain nodes linked with pointers in memory. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Gorman into Harris in order to use tree-based structures instead of list based data structures, to gain the advantages of dynamic memory allocation, which were well-known in the art at the time of the invention.

Regarding dependent claim 6, Matthews teaches VML will use existing HTML mechanisms (p. 7, #3), which includes web browsers. It would have been obvious to have a browser create the data structure, since the browser would have been processing the expressions, and therefore must create the structure.

Regarding dependent claim 7, Mathews teaches the markup language is HTML (p. 7, #3).

Regarding dependent claim 8, Mathews teaches up to 8 adjust values may used (p.25, 2nd para.). Therefore two scalars can map to an expression. Using pointers would have been obvious to one of ordinary skill in the art at the time of the invention as set forth in claim 5 above.

Regarding dependent claim 9, Mathews teaches mapping a single scalar to two expressions (p. 5 equations @0 and @1). Using pointers would have been obvious to one of ordinary skill in the art at the time of the invention as set forth in claim 5 above.

Regarding dependent claim 10, Mathews teaches mapping one expression to two different expressions (p. 25). Using pointers would have been obvious to one of ordinary skill in the art at the time of the invention as set forth in claim 5 above.

Regarding dependent claim 11, Mathews teaches mapping dependent/dependencies relationships between expressions. Using pointers would have been obvious to one of ordinary skill in the art at the time of the invention as set forth in claim 5 above.

Regarding dependent claim 12, Mathews teaches several dependency relationships, and the section (p.22-25) teaches that all levels of dependency are taught including a least dependent and a most dependent. Mathews is silent on how execution will be handled. Harris teaches expressions can have any number of dependencies (col. 1, ll. 56-64), which includes have an expression dependent on an expression dependent on a scalar. Harris also teaches recalculating in an order such that dependent expressions will not be recalculated until its dependent expressions are recalculated first (col. 1, ll. 26-32). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22). Gorman discloses a tree for doing such recalculations (col. 9, ll. 3-5). Trees were well known to contain nodes linked with pointers in memory. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Gorman into Harris in order to use tree-based structures instead of list

based data structures, to gain the advantages of dynamic memory allocation, which were well-known in the art at the time of the invention.

Regarding dependent claim 13, Mathews teaches several dependency relationships, and the section (p.22-25) teaches that all levels of dependency are taught including a least dependent and a most dependent. Mathews is silent on how execution will be handled. Harris teaches expressions can have any number of dependencies (col. 1, ll. 56-64), which includes have an expression dependent on an expression dependent on a scalar, and a third expression. Harris also teaches recalculating in an order such that dependent expressions will not be recalculated until its dependent expressions are recalculated first (col. 1, ll. 26-32). Gorman discloses a tree for doing such recalculations (col. 9, ll. 3-5). Trees were well known to contain nodes linked with pointers in memory. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Gorman into Harris in order to use tree-based structures instead of list based data structures, to gain the advantages of dynamic memory allocation, which were well-known in the art at the time of the invention.

Regarding independent claim 14, Mathews teaches scalars and expressions of scalars that are used to program formatting instructions as set forth in claim 1 above. Mathews does not explicitly teach how the calculations are done. Harris discloses a first object that is a scalar and a second object that is a function of the first (col. 4, ll. 50-59). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be

recalculated (Harris, col. 1, line 60 – col. 2, line 22). Harris does provide a method of mapping dependencies (col. 1, ll. 20-25), but does not explicitly disclose a dependency graph. Garman teaches using a tree (col. 9, ll. 3-5), equivalent to a dependency graph for recalculations. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Garman with Harris and Matthews, as trees were a common method of mapping dependencies (col. 9, ll. 3-5).

Regarding dependent claim 15, Matthew does not teach a recalculation method. Harris discloses changing a dirty bit when the properties change (col. 5, ll. 51-64). Harris teaches adding dependent cells to a recalculation list (col. 7, ll. 60-65), equivalent to propagating the dirty bit. Finally Harris teaches recalculating cells marked for recalculation (col. 8, ll. 24-27). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22).

Regarding dependent claim 16, Matthew does not teach a recalculation method. Harris teaches recalculating at the completion of propagation (col. 8, ll. 16-27). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22).

Regarding dependent claim 17, Matthews does not teach a recalculation method. Harris teaches recalculating in an order such that dependent expressions will not be recalculated until its dependent expressions are recalculated first (col. 1, ll. 26-32). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4). Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22). Harris does provide a method of mapping dependencies (col. 1, ll. 20-25), but does not explicitly disclose a dependency graph. Garman teaches using a tree, equivalent to a dependency graph for recalculations. A tree inherently has a plurality of nodes. Also a most-dependent node would inherently have no dependent nodes, and similarly a least dependent expression, would only be dependent on a scalars. It would have been obvious to modify Garman into Harris and Mathews to enable use of tree-structure rather then the lists of Harris, as trees were a common method of mapping dependencies (col. 9, ll. 3-5).

Regarding dependent claim 18, Mathews teaches that the user interface must be updated upon the changing of values p. 8, #4).

Regarding dependent claim 20, Mathews teaches scalars and expressions of scalars that are used to program formatting instructions as set forth in claim 1 above. Harris teaches adding dependent cells to a recalculation list (col. 7, ll. 60-65), equivalent to propagating the dirty bit. Finally Harris teaches executing cells marked for recalculation (col. 8, ll. 24-27). It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Harris and Mathews, as Matthews's implementation requires a recalculation engine (p. 8, #4).

Additionally, while Harris specifically deals with spreadsheets, the spreadsheet element it deals with is the cells, which are effectively the formulas of Mathews, in that they are formulas that need to be recalculated (Harris, col. 1, line 60 – col. 2, line 22). Harris does provide a method of mapping dependencies (col. 1, ll. 20-25), but does not explicitly disclose a dependency graph. Garman teaches using a tree (col. 9, ll. 3-5), equivalent to a dependency graph for recalculations. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Garman with Harris and Matthews, as trees were a common method of mapping dependencies (col. 9, ll. 3-5).

Regarding dependent claim 21, Matthews does not teach a recalculation method. Harris discloses executing the expressions in a sequential order from least-dependent to most-dependent (col. 7, line 39 – col. 8, line 27).

Regarding dependent claim 22, Mathews and Harris do not explicitly disclose input signals. Garman teaches a signal received by the input changes the value of a scalar property (col. 8, ll. 66-67), and generating a changed property notification (col. 9, ll. 1-3). Upon receive the signal the expressions are executed (col. 9, ll. 1-7). It would have been obvious to modify Garman into Harris and Mathews to enable use of tree-structure rather then the lists of Harris, as trees were a common method of mapping dependencies (col. 9, ll. 3-5).

Regarding dependent claims 26-28, the apparatus analogous to the computers of claims 20-22 are rejected under the same rationale.

Regarding claims 29-33, the instructions for executing the methods of claims 14-18 are rejected under the same rationale.

Response to Arguments

7. Applicant's arguments filed 07/09/2004 have been fully considered but they are not persuasive.

Regarding Applicant's remarks on Claims 4, 19, 23-25:

Applicant alleges that Matthews and Harris are not combinable. The Office has given motivation for the combination in all of its claims and submits that it has made its *prima facie* case. Therefore, the Office has met its requirement the Applicant cited from *Ex parte Skinner*, and the burden shifts to the Applicant to prove that this motivation is improper for whatever reason.

In response to applicant's argument that the references are not combinable because Matthews is a programming language and Harris an application, the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

Finally Applicant alleges one programming in VML would not look to a spreadsheet for programming techniques. However, one programming and *implementation* of VML would look to all types of programming techniques to implement a processor for the language. As Matthews's implementation requires a recalculation engine (p. 8, #4), surely one would turn to a spreadsheet application, which relies on calculations and recalculation.

Regarding Applicant's remarks on Claims 5-18, 20-22, and 26-33:

The issues addressed by Applicant have been addressed in the section above.

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Adam M Queler whose telephone number is (571) 272-4140. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather R Herndon can be reached on (571) 272-4136. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AQ

HEATHER R. HERNDON
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100